



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,913	10/24/2003	Ray Y. Lai	5681-66303	9034
58467	7590	06/24/2010		
MHKKG/Oracle (Sun)			EXAMINER	
P.O. BOX 398			WANG, JUE S	
AUSTIN, TX 78767				
			ART UNIT	PAPER NUMBER
			2193	
			NOTIFICATION DATE	DELIVERY MODE
			06/24/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patent_docketing@intprop.com
ptomhkg@gmail.com

Office Action Summary	Application No. 10/692,913	Applicant(s) LAI, RAY Y.	
	Examiner JUE WANG	Art Unit 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 05 March 2010.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-12, 14-31 and 34-92 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-12, 14-31 and 34-92 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 1-12, 14-32, and 34-92 have been examined.

Double Patenting

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 1, 3, 8-10, 14, 17, 45, 47, 52-54, 56, 57, 69, 71, 76-78, 80, and 81 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 20, 23, 26-28, 35, 37, 38, 41, 44-46, 53, and 55 of co-pending application No. 10/642928.

4. The following is a side-by-side comparison between representative claim 20 of co-pending application No. 10/642928 and the representative claim 1 of the instant application.

claim 20 of co-pending application No. 10/642928	claim 1 of the instant application
20. A method, comprising:	1. A system for integrating Web Services with a business system, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to:
generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components of the business system in accordance with one or more integration design patterns,	generate an integrated Web Service architecture comprising a plurality of heterogeneous components of the business system in accordance with one or more integration design patterns;
wherein said generating a vendor-independent Web Services architecture comprises: generating one or more Use Cases for the Web Service;	wherein, to generate an integrated Web Service architecture, the program instructions are further executable by the processor to: generate one or more Use Cases for the integrated Web Service;
generating a high-level architecture for the Web Service, wherein the high-level architecture identifies two or more entities of the Web Service and the relationships and interactions among the entities;	generate a high-level architecture for the integrated Web Service, wherein the high- level architecture identifies two or more entities of the integrated Web Service and the relationships and interactions among the

	entities;
generating a logical architecture for the Web Service according to the use case scenarios, wherein the logical architecture identifies two or more logical components of the Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers; and implementing the Web Service according to the Web Service Architecture.	generate a logical architecture for the integrated Web Service according to the Use Cases, wherein the logical architecture identifies two or more logical components of the integrated Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers.

5. From the comparison above, the conflicting claims are not patentably distinct from each other even though they are not identical because both the present application and co-pending application No. 10/642928 describe a method of generating a web service architecture comprising a plurality of heterogeneous components in accordance with one or more design patterns, including generating one or more Use Cases, generating a high-level architecture for the web service, and generating a logical architecture for the web service. For example, the system of generating a web services disclosed in claim 1 of the present application is an obvious variation of the method for generating a web service as recited in claim 20 of co-pending application No. 10/642928. While claim 20 of co-pending application No. 10/642928 does not specifically recite that the web service is integrated with a business system and the plurality of heterogeneous components are of a business system, it would have been obvious to one of

Art Unit: 2193

ordinary skill in the art that the at the time of the invention that the web service would be integrated with a business system since the web service is for a business service as recited in claims 26 and 27 of co-pending application No. 10/462928 and a web service providing business services would be integrated with a business system providing the business service. In addition, claim 8 of the instant application is an obvious variation of claim 23 of co-pending application claim 9 of the instant application is an obvious variation of claim 26 of co-pending application No. 10/642928, claim 10 of the instant application is an obvious variation of claim 27 of co-pending application No. 10/642928, claim 14 of the instant application is an obvious variation of claim 35 of co-pending application No. 10/642928, claim 17 of the instant application is an obvious variation of claim 28 of co-pending application No. 10/642928.

6. These are provisional obviousness-type double patenting rejections because the conflicting claims have not in fact been patented.

7. Claims 18-28, 30, and 58-68 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 3, 6, 7, 9, 15, 16, 18-23, 25, 32-34, 36, 41, 42, 44-49, 51, 54-56, 58, 63, 64, 66-71, and 73 of U.S. Patent No. 7,698,398 B1.

8. Although the conflicting claims are not identical, they are not patentably distinct from each other because both the present application and U.S. Patent No. 7,698,398 B1 describe a system and method for generating an web service. For example, claim 18 of the present application corresponds to claim 18 of U.S. Patent No. 7,698,398, where parent claim 1 of U.S.

Art Unit: 2193

Patent No. 7,698,398 recites the identify, translate, categorize, organize, apply, and provide output steps, and claim 18 of U.S. Patent No. 7,698,398 recite defining a plurality of integration tiers defining how the plurality of integration tiers communicate with others of the plurality of integration tiers. While claim 18 of U.S. Patent No. 7,698,398 does not specifically recite that the web service architecture is for implementing integrated business system, it would have been obvious to one of ordinary skill in the art that the at the time of the invention that the web service architecture would be for implementing integrated business system since claim 23 of U.S. Patent No. 7,698,398 recites programmable instructions for integration and interoperability with the web services architecture for existing business functionality.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

10. Claims 1-3, 5, 6, 8-10, 14-17, 45-47, 49, 50, 52-54, 56, 57, 69-71, 73, 74, 76-78, 80, and 81 are rejected under 35 U.S.C. 103(a) as being unpatentable over Curry et al. (US 2003/0233631 A1, hereinafter Curry), in view of Epionet, “Epiowave”, further in view of Huang et al. “A Web Services-Based Framework for Business Integration Solutions” (hereinafter Huang), further in view of Olsen (US 2004/0243583 A1).

Art Unit: 2193

11. As per claim 1, Curry teaches the invention as claimed, including a system for integrating Web Services with a business system, comprising:

a processor (see [0072], [0076], [0176]; EN: software toolsets such as Epiowave is used to integrate the web service and it is well know in the art that software is executed by a processor); and

a memory comprising program instructions (see [0072], [0076], [0176]; EN: software toolsets such as Epiowave is used to integrate the web service and it is well know in the art that software is stored in memory before being executed by a processor) to generate integrated Web Service architectures for integrating Web Services with business systems (see [0021], [0022], [0024]), wherein, to generate an integrated Web Service architecture for integrating a specific Web Service with a specific business system, the program instructions are executable by the processor to:

generate an integrated Web Service architecture comprising a plurality of heterogeneous components of the business system (see [0021], [0022], [0024]), wherein, to generate the integrated Web Service architecture, the program instructions are further executable by the processor to:

generate one or more Use Cases for the integrated Web Service (see [0080]-[0084]);

generate a high-level architecture for the integrated Web Service, wherein the high-level architecture identifies two or more entities of the integrated Web Service and the relationships and interactions among the entities (see [0097]; EN: the context diagram describes the high level architecture); and

generate a logical architecture for the integrated Web Service according to the Use Cases, wherein the logical architecture identifies two or more logical components of the integrated Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers (see [0055]-[0059]; EN: the framework structure including a number of layers is the logical architecture).

Curry does not explicitly teach implementing a Web Services architecture design service to generate integrated Web Service architectures for integrating Web Services with business systems. However, it would have been obvious to one of ordinary skill in the art at the time of the invention that the system of Curry would include a Web Services architecture design service since the Epiowave toolset used in Curry is capable of planning, prototyping, testing, developing, and deploying web services (see pages 1-2 of Epionet; EN: the Epiowave suite is considered as a Web Services architecture design service).

Curry does not explicitly teach the web service architecture is generated in accordance with one or more integration design patterns.

Huang is cited to teach a method of developing web services based business integration using integration design patterns (see page 18, right column, paragraph 2, page 20, left column, paragraphs 2, 3, right column, paragraphs 1-3).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry such that the web service architecture is generated in accordance with one or more integration design patterns as taught by Huang because design patterns are well known in the art and commonly used by programmers since design patterns provide the benefit of capturing a standard solution to a common programming problem for reuse.

Curry, Epionet, and Huang do not explicitly teach providing output indicating the generated integrated Web Service architecture for integrating the Web Service with the specific business system. However, it would have been obvious to one of ordinary skill in the art at the time of the invention that there would have been an output indicating the generated integrated Web Service architecture because the template is used by developers for customization of enterprise solutions (see [0055] of Curry) and as it is well known in the art that a user must be provided with an indication that a template for a web service exists before that template can be customized (see [0005] of Olsen).

12. As per claim 2, Curry teaches defining a plurality of integration tiers (i.e., logical layers of the application may be each physically separated or may be combined into components that include multiple logical layers, see [0151]), one or more basic components (see [0068], [0148]), and one or more Web Services technologies for integration ([0135], [0136]); and define how each of the plurality of integration tiers communicates with others of the plurality of integration tiers (i.e., the sub-architectures are linked using XML as a standard communication mechanism, see [0056], [0136], [0151]).

13. As per claim 3, Curry teaches wherein the plurality of integration tiers comprises one or more of: a client tier, a presentation tier, a business tier, an integration tier, and a resources tier (see [0016], [0056]-[0059]).

Art Unit: 2193

14. As per claim 5, Curry does not explicitly teach the business system is an Enterprise business system.

Huang teaches web services based integration of business solutions including an Enterprise business system (see page 17, right column, paragraph 2, Figure 2, page 18, right column, paragraph 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry such that the business system is an Enterprise business system as taught by Huang because it is well known in the art that web services are provided for Enterprise business systems.

15. As per claim 6, Curry does not explicitly teach the business system is a Cross-Enterprise business system.

Huang teaches web services based integration of business solutions including a Cross-Enterprise business system (see page 16, right column, paragraph 3, bullet B.1, page 17, right column, paragraph 2, Figure 2, page 18, right column, paragraph 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry such that the business system is a Cross-Enterprise business system as taught by Huang because it is well known in the art that web services are provided for Cross-Enterprise business systems.

16. As per claim 8, Curry does not explicitly teach the integrated Web Service architecture comprises: a service provider configured to provide one or more services on an integrated Web

Art Unit: 2193

Service business system implemented according to the integrated Web Service architecture; and one or more service requesters configured to access the one or more services from the service provider via a network.

Huang teaches the integrated Web Service architecture comprises: a service provider configured to provide one or more services on an integrated Web Service business system implemented according to the integrated Web Service architecture; and one or more service requesters configured to access the one or more services from the service provider via a network (see page 17, Figure 1, left column, paragraphs 1, 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention that the Web Service architecture of Curry would have one or more service providers and one or more service requesters as taught by Huang since service providers and service requesters are part of a web services model well known in the art (see page 17, Figure 1, left column, paragraphs 1, 2 of Huang).

17. As per claim 9, Huang teaches wherein the integrated Web Service business system is a Business-to-Business system, wherein the service provider is a business service provider, and wherein the service requester is an end user (see page 16, right column, last paragraph, bullet B.1).

18. As per claim 10, Huang teaches wherein the integrated Web Service business system is a Business-to-Business system, wherein the service provider is a business service provider, and

Art Unit: 2193

wherein the service requester is a business server (see page 16, right column, last paragraph, bullet B.3).

19. As per claim 14, Huang teaches the integration design patterns include an Application-to-Application design pattern (see page 20, left column, paragraph 1, Fig. 5; EN: the composite design pattern of the service composition is considered as an Application-to-Application pattern).

20. As per claim 15, Huang teaches the integration pattern includes an Open Process integration design pattern (see page 20, left column, paragraph 3, right column, paragraph 1; EN: the Mediator pattern is considered as an Open Process design pattern).

21. As per claim 16, Huang teaches wherein the design patterns include one of a Service Consolidation-Broker integration design pattern (see page 20, left column, first paragraph, bullet e), right column, paragraph 3; EN: the state pattern used in the Adaptive Document brokering service is considered as a Service Consolidation-Broker integration design pattern).

22. As per claim 17, Curry teaches the layers comprises a transport layer for delivering messages between components of the integrated Web Services (see [0136]); and a management layer configured for provisioning of the integrated Web Services and for monitoring and administration of the integrated Web Services (i.e., state/session, and user management functions, see [0028]).

Art Unit: 2193

23. As per claims 45-47, 49, 50, 52-54, 56, and 57, the limitations recited in these method claims are substantially similar to those recited in claims 1-3, 5, 6, 8-10, 14, and 17. Therefore, they are rejected using the same reasons as claims 1-3, 5, 6, 8-10, 14, and 17.

24. As per claims 69-71, 73, 74, 76-78, 80, and 81, the limitations recited in these computer-accessible medium claims are substantially similar to those recited in claims 1-3, 5, 6, 8-10, 14, and 17. Therefore, they are rejected using the same reasons as claims 1-3, 5, 6, 8-10, 14, and 17.

25. Claims 4, 48, 72 are rejected under 35 U.S.C. 103(a) as being unpatentable over Curry et al. (US 2003/0233631 A1, hereinafter Curry), in view of Epionet, “Epiowave”, further in view of Huang et al. “A Web Services-Based Framework for Business Integration Solutions” (hereinafter Huang), further in view of Olsen (US 2004/0243583 A1), further in view of Connell et al. (US 2003/0074401 A1, hereinafter Connell).

26. As per claim 4, Curry, Epionet, Huang, and Olsen do not explicitly teach integration of one or more Enterprise Application Interface (EAI) products with one or more Web Services technologies.

Connell teaches integration of one or more Enterprise Application Interface (EAI) products with one or more Web Services technologies (see [0002]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry, Epionet, Huang, and Olsen to provide integration of one or more Enterprise Application Interface (EAI) products with one or more Web Services technologies as

Art Unit: 2193

taught by Connell to provide communication between heterogeneous computers systems interconnected in a computer network such as EAI software and web services (see [0002] of Connell).

27. As per claims 48 and 72, the limitations recited in these claim are substantially similar to those recited in claim 4. Therefore, they are rejected using the same reasons as claims 4.

28. Claims 7, 11, 51, 55, 75, and 79 are rejected under 35 U.S.C. 103(a) as being unpatentable over Curry et al. (US 2003/0233631 A1, hereinafter Curry), in view of Epionet, “Epiowave”, further in view of Huang et al. “A Web Services-Based Framework for Business Integration Solutions” (hereinafter Huang), further in view of Olsen (US 2004/0243583 A1), further in view of Chappell et al. “Java Web Services” (hereinafter Chappell).

29. As per claim 7, Curry, Epionet, Huang, and Olsen do not explicitly teach wherein the plurality of heterogeneous components of the business system includes one or more legacy mainframe systems.

Chappell teaches web services with one or more legacy mainframe systems (section 2.1, paragraph 3).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry, Epionet, Huang, and Olsen such that the plurality of heterogeneous components of the business system includes one or more legacy mainframe system as taught by

Art Unit: 2193

Chappell such that the web service can provide functions that are provided by legacy systems that already exist (see section 2.1, paragraph 3 of Chappell).

30. As per claims 11 and 12, Huang teaches the design patterns include one or more integration and interoperability design patterns including a asynchronous web service design pattern (i.e., asynchronous transaction support using the command pattern to encapsulate requests, see page 18, right column, paragraph 3, bullet 2, page 20, right column, paragraph 2). Huang does not explicitly teach the design pattern is directed to mainframes, however, it would have been obvious to one of ordinary skill in the art at the time of the invention that the design pattern for transaction could be directed to mainframes since web services are commonly provided for mainframe systems to access functionality provided by the mainframe system (see section 2.1, paragraph 3 of Chappell).

31. As per claims 51 and 75, the limitations recited in these claims are substantially similar to those recited in claim 7. Therefore, they are rejected using the same reasons as claim 7.

32. As per claims 55 and 79, the limitations recited in these claims are substantially similar to those recited in claim 12. Therefore, they are rejected using the same reasons as claim 12.

33. Claims 18-24, 27-29, 58-64, 67-68, 82-88, 91, and 92 are rejected under 35 U.S.C. 103(a) as being unpatentable over Curry et al. (US 2003/0233631 A1, hereinafter Curry), in view of Epionet, "Epiowave", further in view of Siegel, "Using OMG's Model Driven Architecture

Art Unit: 2193

(MDA) to Integrate Web Services”, further in view of Huang et al. “A Web Services-Based Framework for Business Integration Solutions” (hereinafter Huang), further in view of Olsen (US 2004/0243583 A1).

34. As per claim 18, Curry teaches the invention as claimed, including a system for generating an integrated Web Service architecture, comprising:

a processor (see [0072], [0076], [0176]; EN: software toolsets such as Epiowave is used to integrate the web service and it is well know in the art that software is executed by a processor); and

a memory comprising program instructions (see [0072], [0076], [0176]; EN: software toolsets such as Epiowave is used to integrate the web service and it is well know in the art that software is stored in memory before being executed by a processor) to generate integrated Web Service architectures for integrating Web Services with business systems (see [0021], [0022], [0024]), wherein, to generate an integrated Web Service architecture for integrating a specific Web Service with a specific business system, the program instructions are executable by the processor to:

identify one or more components of the integrated Web Service architecture according to one or more use case requirements for the specific integrated Web Service business system (see Fig 2, [0051, [0052], [0074]-[0080]);

determining a plurality of Web Service components for the integrated Web Service architecture, wherein the Web Service components include software components (i.e., prototypes and classes are developed, see [0118], [0119], [0150]);

categorizing the Web Service components into two or more related groups according to a Web Services architecture integration framework (i.e., categorization of components so that a search engine can be used for efficient retrieval, see [0164]).

define a plurality of integration tiers and one or more Web Services technologies according to a Web Services architecture integration framework (i.e., logical layers of the application may be each physically separated or may be combined into components that include multiple logical layers, see [0056]-[0059], [0151]);

define how each of the plurality of integration tiers communicates with others of the plurality of integration tiers in the integrated Web Service architecture according to the Web Services architecture integration framework (i.e., the sub-architectures are linked using XML as a standard communication mechanism, see [0056], [0136], [0151]);

organize the groups of Web Service components according to the plurality of integration tiers and two or more layers of the integrated Web Service architecture (see [0057]-[0059], [0151]);

Curry does not explicitly teach implementing a Web Services architecture design service to generate integrated Web Service architectures for integrating Web Services with business systems. However, it would have been obvious to one of ordinary skill in the art at the time of the invention that the system of Curry would include a Web Services architecture design service since the Epiowave toolset used in Curry is capable of planning, prototyping, testing, developing, and deploying web services (see pages 1-2 of Epionet; EN: the Epiowave suite is considered as a Web Services architecture design service).

Curry and Epionet do not teach translate the one or more use case requirements for the specific integrated Web Service business system and one or more technical constraints for the specific integrated Web Service business system to determine a plurality of Web Service components for the integrated Web Service architecture, wherein the Web Service components include software components.

Siegel teaches a method of integrating web services with business systems, including translating one or more use case requirements for the specific Web Service business system and one or more technical constraints for the specific integrated Web Service business system to determine a plurality of Web Service components for the integrated Web Service architecture, wherein the Web Service components include software components (i.e., MDA tools generate interface definitions, application code, makefiles, and configuration files for the PSM's middleware platform, see page 5, last paragraph, page 6, paragraphs 1-3, page 8, last paragraph).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry and Epionet to translate the one or more use case requirements for the specific integrated Web Service business system and one or more technical constraints for the specific integrated Web Service business system to determine a plurality of Web Service components for the integrated Web Service architecture as taught by Siegel such that tools can automate and thereby simplify most of the building of distributed applications (see page 1, paragraph 3 of Siegel).

Curry does not explicitly teach applying one or more design patterns to the integrated Web Service architecture where appropriate.

Huang is cited to teach a method of developing web services based business integration using integration design patterns (see page 18, right column, paragraph 2, page 20, left column, paragraphs 2, 3, right column, paragraphs 1-3).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry such that the web service architecture is generated in accordance with one or more integration design patterns as taught by Huang because design patterns are well known in the art and commonly used by programmers since design patterns provide the benefit of capturing a standard solution to a common programming problem for reuse.

Curry, Epionet, Siegel, and Huang do not explicitly teach providing output indicating the generated integrated Web Service architecture for integrating the Web Service with the specific business system. However, it would have been obvious to one of ordinary skill in the art at the time of the invention that there would have been an output indicating the generated integrated as Web Service architecture because the template is used by developers for customization of enterprise solutions (see [0055] of Curry) and as it is well known in the art that a user must be provided with an indication that a template for a web service exists before that template can be customized (see [0005] of Olsen).

35. As per claim 19, Curry does not explicitly teach the integrated Web Service architecture comprises: a service provider configured to provide one or more services on an integrated Web Service business system implemented according to the integrated Web Service architecture; and one or more service requesters configured to access the one or more services from the service provider via a network.

Huang teaches the integrated Web Service architecture comprises: a service provider configured to provide one or more services on an integrated Web Service business system implemented according to the integrated Web Service architecture; and one or more service requesters configured to access the one or more services from the service provider via a network (see page 17, Figure 1, left column, paragraphs 1, 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention that the Web Service architecture of Curry would have one or more service providers and one or more service requesters as taught by Huang since service providers and service requesters are part of a web services model well known in the art (see page 17, Figure 1, left column, paragraphs 1, 2 of Huang).

36. As per claim 20, Huang teaches wherein the integrated Web Service business system is a Business-to-Business system, wherein the service provider is a business service provider, and wherein the service requester is an end user (see page 16, right column, last paragraph, bullet B.1).

37. As per claim 21, Huang teaches wherein the integrated Web Service business system is a Business-to-Business system, wherein the service provider is a business service provider, and wherein the service requester is a business server (see page 16, right column, last paragraph, bullet B.3).

Art Unit: 2193

38. As per claim 22, Curry teaches the layers comprises a transport layer for delivering messages between components of the integrated Web Services (see [0136]); and a management layer configured for provisioning of the integrated Web Services and for monitoring and administration of the integrated Web Services (i.e., state/session, and user management functions, see [0028]).

39. As per claim 23, Curry does not explicitly teach the specific integrated Web Service business system is an Enterprise business system.

Huang teaches web services based integration of business solutions including an Enterprise business system (see page 17, right column, paragraph 2, Figure 2, page 18, right column, paragraph 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry such that the business system is an Enterprise business system as taught by Huang because it is well known in the art that web services are provided for Enterprise business systems.

40. As per claim 24, Curry does not explicitly teach the specific integrated Web Service business system is a Cross-Enterprise business system.

Huang teaches web services based integration of business solutions including a Cross-Enterprise business system (see page 16, right column, paragraph 3, bullet B.1, page 17, right column, paragraph 2, Figure 2, page 18, right column, paragraph 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry such that the business system is a Cross-Enterprise business system as taught by Huang because it is well known in the art that web services are provided for Cross-Enterprise business systems.

41. As per claim 27, Curry teaches wherein the plurality of integration tiers comprises one or more of: a client tier, a presentation tier, a business tier, an integration tier, and a resources tier (see [0016], [0056]-[0059]).

42. As per claim 28, Huang teaches the integration design patterns include an Application-to-Application design pattern (see page 20, left column, paragraph 1, Fig. 5; EN: the composite design pattern of the service composition is considered as an Application-to-Application pattern).

43. As per claim 29, Huang teaches the integration pattern includes an Open Process integration design pattern (see page 20, left column, paragraph 3, right column, paragraph 1; EN: the Mediator pattern is considered as an Open Process design pattern).

44. As per claims 58-64 and 67-68, the limitations recited in these method claims are substantially similar to those recited in claims 18-24 and 27-28. Therefore, they are rejected using the same reasons as claims 18-24 and 27-28.

Art Unit: 2193

45. As per claims 82-88, 91, and 92, the limitations recited in these computer-accessible medium claims are substantially similar to those recited in claims 18-24, 27, and 28. Therefore, they are rejected using the same reasons as claims 18-24, 27, and 28.

46. Claims 25, 30, 65, and 89 are rejected under 35 U.S.C. 103(a) as being unpatentable over Curry et al. (US 2003/0233631 A1, hereinafter Curry), in view of Epionet, “Epiowave”, further in view of Siegel. “Using OMG’s Model Driven Architecture (MDA) to Integrate Web Services”, further in view of Huang et al. “A Web Services-Based Framework for Business Integration Solutions” (hereinafter Huang), further in view of Olsen (US 2004/0243583 A1), further in view of Chappell et al. “Java Web Services” (hereinafter Chappell).

47. As per claim 25, Curry, Epionet, Siegel, Huang, and Olsen do not explicitly teach wherein the plurality of heterogeneous components of the business system includes one or more legacy mainframe systems.

Chappell teaches web services with one or more legacy mainframe systems (section 2.1, paragraph 3).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry, Epionet, Siegel, Huang, and Olsen such that the plurality of heterogeneous components of the business system includes one or more legacy mainframe system as taught by Chappell such that the web service can provide functions that are provided by legacy systems that already exist (see section 2.1, paragraph 3 of Chappell).

48. As per claim 30, Huang teaches the design patterns include one or more integration and interoperability design patterns including a asynchronous web service design pattern (i.e., asynchronous transaction support using the command pattern to encapsulate requests, see page 18, right column, paragraph 3, bullet 2, page 20, right column, paragraph 2). Huang does not explicitly teach the design pattern is directed to mainframes, however, it would have been obvious to one of ordinary skill in the art at the time of the invention that the design pattern for transaction could be directed to mainframes since web services are commonly provided for mainframe systems to access functionality provided by the mainframe system (see section 2.1, paragraph 3 of Chappell).

49. As per claims 65 and 89, the limitations recited in these claims are substantially similar to those recited in claim 25. Therefore, they are rejected using the same reasons as claim 25.

50. Claims 26, 66, and 90 are rejected under 35 U.S.C. 103(a) as being unpatentable over Curry et al. (US 2003/0233631 A1, hereinafter Curry), in view of Epionet, “Epiowave”, further in view of Siegel. “Using OMG’s Model Driven Architecture (MDA) to Integrate Web Services”, further in view of Huang et al. “A Web Services-Based Framework for Business Integration Solutions” (hereinafter Huang), further in view of Olsen (US 2004/0243583 A1), further in view of Connell et al. (US 2003/0074401 A1, hereinafter Connell).

Art Unit: 2193

51. As per claim 26, Curry, Epionet, Siegel, Huang, and Olsen do not explicitly teach integration of one or more Enterprise Application Interface (EAI) products with one or more Web Services technologies.

Connell teaches integration of one or more Enterprise Application Interface (EAI) products with one or more Web Services technologies (see [0002]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry, Epionet, Siegel, Huang, and Olsen to provide integration of one or more Enterprise Application Interface (EAI) products with one or more Web Services technologies as taught by Connell to provide communication between heterogeneous computers systems interconnected in a computer network such as EAI software and web services (see [0002] of Connell).

52. As per claims 66 and 90, the limitations recited in these claims are substantially similar to those recited in claim 26. Therefore, they are rejected using the same reasons as claim 26.

53. Claims 31, 34, 35, 37-39, and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Curry et al. (US 2003/0233631 A1, hereinafter Curry), in view of Curtis et al. (US 2003/0115377 A1, hereinafter Curtis), further in view of Epionet, “Epiowave”, further in view of Huang et al. “A Web Services-Based Framework for Business Integration Solutions” (hereinafter Huang).

Art Unit: 2193

54. As per claim 31, Curry teaches the invention as claimed, including an integrated Web Services business system, comprising:

one or more computers configured to implement (see [0072], [0076], [0176]; EN: software toolsets such as Epiowave is used to integrate the web service and it is well know in the art that software is executed by a computer):

a plurality of heterogeneous business components organized according to an integrated Web Service architecture for the integrated Web Services business system (see [0024], [0151]);

a plurality of tiers implemented according to the integrated Web Service architecture for the integrated Web Services business system, wherein the plurality of integration tiers comprises a presentation tier, a business tier, and a resources tier (i.e., automating layer separability, see [0016], [0028]);

a Web Service comprising one or more Web Services technologies defined by the integrated Web Service architecture for the integrated Web Services business system, wherein the Web Service is configured to provide interoperability among the plurality of heterogeneous business components via a network according to the integrated Web Service architecture for the integrated Web Services business system (i.e., web service applications are created by integrating the classes and components according to the business object framework, see [0024]);

wherein the integrated Web Service architecture for the integrated Web Services business system is configured according to a vendor-independent architecture framework for integrating Web Services technologies with business systems comprising a plurality of heterogeneous components in accordance with a structured integration methodology (i.e., a pre-built

Art Unit: 2193

architecture that allows developers to rapidly create applications based on business components and web services, see [0055]-[0059]).

Curry does not explicitly wherein the plurality of integration tiers also comprises a client tier and an integration tier.

Curtis is cited to teach a method for separating an integrated management architecture into tiers, including a client tier, a presentation tier, a business tier, an integration tier, and a resources tier (see [0007], [0028]-[0033]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry such that the plurality of tiers also include a client tier and an integration tier because the n-tier architecture for web services is well known in the art (see [0016] of Curry) and separating web services into a tier structure that including a client tier and an integration tier is also well known (see [0007] of Curtis), therefore, including the client and integration tier is a design choice based on the requirements of the application.

Curry does not explicitly teach a computer-implemented Web Services business system architecture design service to generate integrated Web Service architectures for integrating Web Services with business systems. However, it would have been obvious to one of ordinary skill in the art at the time of the invention that the system of Curry would include a Web Services architecture design service since the Epiowave toolset used in Curry is capable of planning, prototyping, testing, developing, and deploying web services (see pages 1-2 of Epionet; EN: the Epiowave suite is considered as a Web Services architecture design service).

Curry does not explicitly teach that the integrated web services business system is configured according to one or more design patterns.

Huang is cited to teach a method of developing web services based business integration using integration design patterns (see page 18, right column, paragraph 2, page 20, left column, paragraphs 2, 3, right column, paragraphs 1-3).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry such that the web service architecture is generated in accordance with one or more integration design patterns as taught by Huang because design patterns are well known in the art and commonly used by programmers since design patterns provide the benefit of capturing a standard solution to a common programming problem for reuse.

55. As per claim 34, Curry does not explicitly teach the business system is an Enterprise business system.

Huang teaches web services based integration of business solutions including an Enterprise business system (see page 17, right column, paragraph 2, Figure 2, page 18, right column, paragraph 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry such that the business system is an Enterprise business system as taught by Huang because it is well known in the art that web services are provided for Enterprise business systems.

56. As per claim 35, Curry does not explicitly teach the business system is a Cross-Enterprise business system.

Huang teaches web services based integration of business solutions including a Cross-Enterprise business system (see page 16, right column, paragraph 3, bullet B.1, page 17, right column, paragraph 2, Figure 2, page 18, right column, paragraph 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry such that the business system is a Cross-Enterprise business system as taught by Huang because it is well known in the art that web services are provided for Cross-Enterprise business systems.

57. As per claim 37, Curry does not explicitly teach the integrated Web Service architecture comprises: a service provider configured to provide one or more services on an integrated Web Service business system implemented according to the integrated Web Service architecture; and one or more service requesters configured to access the one or more services from the service provider via a network.

Huang teaches the integrated Web Service architecture comprises: a service provider configured to provide one or more services on an integrated Web Service business system implemented according to the integrated Web Service architecture; and one or more service requesters configured to access the one or more services from the service provider via a network (see page 17, Figure 1, left column, paragraphs 1, 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention that the Web Service architecture of Curry would have one or more service providers and one or more service requesters as taught by Huang since service providers and service requesters are

Art Unit: 2193

part of a web services model well known in the art (see page 17, Figure 1, left column, paragraphs 1, 2 of Huang).

58. As per claim 38, Huang teaches wherein the integrated Web Service business system is a Business-to-Business system, wherein the service provider is a business service provider, and wherein the service requester is an end user see page 16, right column, last paragraph, bullet B.1).

59. As per claim 39, Huang teaches wherein the integrated Web Service business system is a Business-to-Business system, wherein the service provider is a business service provider, and wherein the service requester is a business server (see page 16, right column, last paragraph, bullet B.3).

60. As per claim 41, Huang teaches the design patterns include one or more integration design patterns (see page 20, left column, paragraphs 2, 3, right column, paragraphs 1-3).

61. Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Curry et al. (US 2003/0233631 A1, hereinafter Curry), in view of Curtis et al. (US 2003/0115377 A1, hereinafter Curtis), further in view of Epionet, “Epiowave”, further in view of Huang et al. “A Web Services-Based Framework for Business Integration Solutions” (hereinafter Huang), further in view of Connell et al. (US 2003/0074401 A1, hereinafter Connell).

Art Unit: 2193

62. As per claim 32, Curry, Curtis, Epionet, and Huang do not explicitly teach integration of one or more Enterprise Application Interface (EAI) products with one or more Web Services technologies.

Connell teaches integration of one or more Enterprise Application Interface (EAI) products with one or more Web Services technologies (see [0002]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry, Curtis, Epionet, and Huang to provide integration of one or more Enterprise Application Interface (EAI) products with one or more Web Services technologies as taught by Connell to provide communication between heterogeneous computers systems interconnected in a computer network such as EAI software and web services (see [0002]0 of Connell).

63. Claims 36 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Curry et al. (US 2003/0233631 A1, hereinafter Curry), in view of Curtis et al. (US 2003/0115377 A1, hereinafter Curtis), further in view of Epionet, “Epiowave”, further in view of Huang et al. “A Web Services-Based Framework for Business Integration Solutions” (hereinafter Huang), further in view of Chappell et al. “Java Web Services” (hereinafter Chappell).

64. As per claim 36, Curry, Curtis, Epionet, and Huang do not explicitly teach wherein the plurality of heterogeneous components of the business system includes one or more legacy mainframe systems.

Chappell teaches web services with one or more legacy mainframe systems (section 2.1, paragraph 3).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry, Curtis, Epionet, and Huang such that the plurality of heterogeneous components of the business system includes one or more legacy mainframe system as taught by Chappell such that the web service can provide functions that are provided by legacy systems that already exist (see section 2.1, paragraph 3 of Chappell).

65. As per claims 40, Huang teaches the design patterns include one or more integration and interoperability design patterns including a asynchronous web service design pattern (i.e., asynchronous transaction support using the command pattern to encapsulate requests, see page 18, right column, paragraph 3, bullet 2, page 20, right column, paragraph 2). Huang does not explicitly teach the design pattern is directed to mainframes, however, it would have been obvious to one of ordinary skill in the art at the time of the invention that the design pattern for transaction could be directed to mainframes since web services are commonly provided for mainframe systems to access functionality provided by the mainframe system (see section 2.1, paragraph 3 of Chappell).

66. Claims 42 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Curry et al. (US 2003/0233631 A1, hereinafter Curry), in view of Huang et al. "A Web Services-Based Framework for Business Integration Solutions" (hereinafter Huang), further in view of Olsen (US 2004/0243583 A1).

67. As per claim 42, Curry is cited to teach the invention as claimed, including a system integrating Web Services with a business system, comprising:

computer-implemented means for generating an integrated Web Services architecture for integrating a Web Service with a business system (see [0055]; EN: the pre-built architecture is a Web Services architecture);

computer-implemented means for applying a Web Services structured methodology to the generated integrated Web Service architecture to identify heterogeneous components for the integrated Web Service architecture and to organize the heterogeneous components according to the integrated Web Service architecture (see [0055]-[0059], [0104]-[0107]); and

wherein said computer-implemented means for applying a Web Services structured methodology to the generated integrated Web Service architecture comprises means for providing integration and interoperability with the integrated Web Service architecture for existing business functionality of the business system (see [0058]);

computer-implemented means for providing output indicating the generated integrated Web Service architecture for integrating the Web Service with the business system (i.e., a prototype is defined to demonstrate functions to a customer or end user, see [0110]; EN: the output is the packaged web service); and

computer-implemented means for implementing the integrated Web Service comprising the identified heterogeneous components organized according to the integrated Web Service architecture (see [0024]-[0027], [0062]-[0071]).

Curry does not explicitly teach that the integrated web services business system is configured according to one or more design patterns.

Huang is cited to teach a method of developing web services based business integration using integration design patterns (see page 18, right column, paragraph 2, page 20, left column, paragraphs 2, 3, right column, paragraphs 1-3).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry such that the web service architecture is generated in accordance with one or more integration design patterns as taught by Huang because design patterns are well known in the art and commonly used by programmers since design patterns provide the benefit of capturing a standard solution to a common programming problem for reuse.

Curry and Huang do not explicitly teach computer implemented means for providing output indicating the generated integrated Web Service architecture for integrating the Web Service with the specific business system. However, it would have been obvious to one of ordinary skill in the art at the time of the invention that there would have been an output indicating the generated integrated Web Service architecture because the template is used by developers for customization of enterprise solutions (see [0055] of Curry) and as it is well known in the art that a user must be provided with an indication that a template for a web service exists before that template can be customized (see [0005] of Olsen).

68. As per claim 43, Curry does not explicitly teach the business system is one of an Enterprise business system and a Cross-Enterprise business system.

Art Unit: 2193

Huang teaches web services based integration of business solutions including an Enterprise business system (see page 17, right column, paragraph 2, Figure 2, page 18, right column, paragraph 2) and a Cross-Enterprise business system (see page 16, right column, paragraph 3, bullet B.1, page 17, right column, paragraph 2, Figure 2, page 18, right column, paragraph 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry such that the business system is an Enterprise business system as taught by Huang because it is well known in the art that web services are provided for Enterprise business systems and Cross-Enterprise business system.

69. Claim 44 is rejected under 35 U.S.C. 103(a) as being unpatentable over Curry et al. (US 2003/0233631 A1, hereinafter Curry), in view of Huang et al. "A Web Services-Based Framework for Business Integration Solutions" (hereinafter Huang), further in view of Olsen (US 2004/0243583 A1), further in view of Chappell et al. "Java Web Services" (hereinafter Chappell).

70. As per claim 44, Curry, Huang, and Olsen do not explicitly teach wherein the plurality of heterogeneous components of the business system includes one or more legacy mainframe systems.

Chappell teaches web services with one or more legacy mainframe systems (section 2.1, paragraph 3).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Curry, Huang, and Olsen such that the plurality of heterogeneous components of the business system includes one or more legacy mainframe system as taught by Chappell such that the web service can provide functions that are provided by legacy systems that already exist (see section 2.1, paragraph 3 of Chappell).

Response to Arguments

71. Rejection of claims under §103(a):

72. As per independent claim 1, Applicants argued that Curry does not teach the invention as claimed because Curry fails to teach program instructions executable by a processor to perform the steps as recited in the limitation. Applicant argued that Curry does not describe a computer system that implements the method illustrated in Fig 1. At most, Curry describes that particular steps or sub-steps of Curry's method may be assisted by or performed using existing software tools. Applicant's arguments have been fully considered and Examiner respectfully disagrees. Examiner submits that the Epiowave software described in Curry would provide the programmable instructions to perform the steps (Examiner has further cited the new reference Epionet which gives more detail about the Epiowave software used in Curry, specifically, Epionet teaches the Epiowave suite contains EpioDesigner that automates the process of planning, prototyping, and testing of Enterprise Web systems, and EpioBuilder that provides a set of tools and services to develop Enterprise web systems). Examiner interprets the programmable instructions to generate the use cases, high-level architectures, logical structure etc. as programmable instructions that generate the use cases, high-level architectures etc. based

Art Unit: 2193

on user input, so a user may design the use case, high-level architecture, but the programmable instructions are used to generate a digital representation of the artifacts. This interpretation is supported by the specification. For example, page 243, lines 21-23 of the specification states “architects collect user requirement and encapsulate them into use case”. As such, Examiner believes that Curry meets these limitations because even though user input is required for designing the use cases, context diagrams etc., the ultimate product is a software artifact in digital format (see [0172]) and modeling software tools, code development software tools, and deployment tools are used to model, develop, and deploy the application. Applicant also argued that Curry teaches a standard framework structure that clearly pre-exists because the framework is prebuilt. In response, Examiner submits that the pre-built framework is used as a blueprint or generic application based upon which the specific application is generated (see [0055]). The framework template is a software artifact and therefore it is generated by a computer. Even though the framework template is prebuilt, customization still needs to be added to the framework template for the specific web service under development (see [0055]) and therefore, a specific instance of the framework template must be generated for the web service under development since modifying the original would change the template and render it unusable for subsequent development for different web services.

73. As per claim 1, Applicant also argued that the office fail to establish a proper prima facie reason to combine Curry and Huang because it is unclear how Curry’s system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry’s “Web Service development method” as disclosed.

Art Unit: 2193

Curry and Huang each propose distinctly different methods for achieving similar results.

Modifying Curry with Huang's specific "design patterns" would appear to necessitate significant changes in Curry's system. In response, it is not clear to the Examiner why incorporating the use of design patterns in the development method taught by Curry would necessitate significant change as claimed by the Applicant. It is well known in the art that design patterns provide a template for solving a problem and software engineers widely use design patterns to solve problems following that template. Therefore, Examiner believes that it is obvious that one of ordinary skill in the art such as a software engineer, developing code for a particular problem (i.e., web service with business integration) would have the necessary skills to use design patterns for that particular problem.

74. As per independent claims 18, 31, and 42, the arguments presented are similar to those presented for claim 1. Therefore, the arguments are not persuasive for the same reasons as those presented for claim 1.

Conclusion

75. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Roberts et al. (US 6,560,633 B1) is cited to teach a method for creating network services by transforming an xml runtime model in response to an iterative input process.
- Fisher (US 2003/0188291 A1) is cited to teach design and redesign of enterprise applications.

Art Unit: 2193

- Epionet, “EpioBuilder Framework”, June 2002, retrieved from:
http://web.archive.org/web/20020613050233/epionet.com/products/epiobuilder_framework.html
- Epionet, “EpioBuilder” June 2002, retrieved from:
<http://web.archive.org/web/20020604030631/epionet.com/products/epiobuilder.html>
- Epionet, “EpioDesigner”, June 2002, retrieved from:
<http://web.archive.org/web/20020604031025/epionet.com/products/epiodesigner.html>
- Epionet, “Epio Business Server” June 2002, retrieved from:
<http://web.archive.org/web/20020604030302/epionet.com/products/epiobusinessserver.html>

76. Applicant’s amendment necessitated the new ground(s) of rejection presented in this office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP §706.07(a).

Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

77. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jue S. Wang whose telephone number is (571) 270-1655. The examiner can normally be reached on M-Th 7:30 am - 5:00pm (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Lewis A. Bullock, Jr./
Supervisory Patent Examiner, Art Unit 2193

Jue Wang
Examiner
Art Unit 2193